

*Amendments in the Claims*

1. (currently amended) A method for identifying user interface (UI) objects in a markup-language stream, the method comprising the steps of:

receiving a predefined grammar;

A) automatically generating building a parser computer program based on the predefined grammar using an automated parser generator tool that accepts a source input file containing a predefined grammar, the parser computer program configured for;

B) scanning any of (i) the markup-language stream and (ii) a corresponding document object model (DOM) to generate tokens; and

C) parsing the tokens using the parser to identify one or more UI objects.

2. (original) The method of claim 1, wherein said markup-language stream drives a markup-language-based browser application, and wherein the scanning step includes scanning the DOM generated by a browser that displays that application.

3. (original) The method of claim 1, wherein the scanning step includes identifying elements of the DOM by traversal thereof.

4. (original) The method of claim 3, wherein the grammar is application-specific.

5. (original) The method of claim 3, wherein the scanning step includes generating one or more tokens for each parsed DOM element.

6. (original) The method of claim 3, wherein scanning step includes mapping DOM elements to tokens.

7. (currently amended) The method of claim 1, wherein the parser computer program parses the tokens according to the grammar to identify and distinguish among UI objects in the markup-language stream.

8. (original) The method of claim 7, wherein said UI objects comprise user input fields, text fields, metatags, unprintable markup-language, and in-line images.

9. (original) The method of claim 1, wherein the scanning and parsing steps are adapted to identify UI objects that correspond to elements displayed in the markup-language application.

10. (currently amended) The method of claim 9, wherein ~~said~~ the parser computer program is operable to groups the tokens into syntactic structures that identify items displayed by the markup-language application.

11. (currently amended) The method of claim 9, wherein said step of scanning ~~can include~~ comprises identifying similarly formatted markup-language elements based on their markup-language attributes such as classname, font size, style, tag color, and size.

12. (original) The method of claim 9, wherein said objects comprise name, content, shape, location, and properties.

13. (currently amended) The method of claim 1, wherein automatically generating said the parser computer program comprises executing YACC (“Yet Another Compiler-Compiler”).

14. (cancelled).

15. (cancelled). ~~The method of claim 13, wherein said parser is built manually by hand-programming.~~

16. (previously presented) The method of claim 1, wherein the parser is a LALR(1) parser.

17. (previously presented) The method of claim 1, wherein the parser is a LR(1) parser.

18. (original) The method of any of claims 1 - 17, wherein the markup language is any of HTML, XHTML and XUL.

19. (currently amended) A digital data processing system for identifying user interface (UI) objects in markup-language-based applications comprising:

~~—— a digital data processing that executes a scanner which receives markup-language DOM and generating one or more tokens for each DOM element, and~~

~~—— the digital data processor further executing a parser that is coupled to the scanner and that receives tokens, and parsing said tokens based on a grammar, and generating a list of UI objects~~

an parser computer program generator operable to generate a parser computer program based on a predefined grammar using an automated parser generator tool, the parser computer program operable to:

scan any of (i) a markup-language stream and (ii) a corresponding document object model (DOM) to generate tokens; and

parse the tokens to generate a list of UI objects.

20. (original) The system of claim 19, wherein the list of UI objects corresponds to elements displayed by the markup-language DOM.

21. (original) The system of claim 20, wherein said UI objects comprise name, content, shape, location, and properties.

22. (original) The system of claim 19, wherein the grammar is application-specific.

23. (original) The system of claim 19, wherein said tokens are interpreted according to the grammar to identify and distinguish among UI objects of a markup-language application's display.

24. (original) The system of claim 23, wherein the UI objects comprise user input fields, text fields, metatags, unprintable markup-language, and in-line images.

25. (currently amended) The system of any of claims ~~18~~ 19 - 24, wherein the markup language is any of HTML, XHTML and XUL.